UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/802,857	03/12/2001	Akihiko Koh	SON-2047	3304

23353 7590 06/29/2010

RADER FISHMAN & GRAUER PLLC
LION BUILDING
1233 20TH STREET N.W., SUITE 501
WASHINGTON, DC 20036

EXAMINER	
YIGDALL, MICHAEL J	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
06/29/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/802,857	KOH ET AL.	
	Examiner	Art Unit	
	Michael J. Yigdall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 June 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *27,28,40 and 45-52* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *27,28,40 and 45-52* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *02/26/2010*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. The decision of the Board of Patent Appeals and Interferences mailed on May 14, 2010 presented a new ground of rejection within the provisions of 37 C.F.R. § 41.50(b). Applicant's reply filed on June 4, 2010 reopens prosecution. Claims 27, 28, 40 and 45-52 are pending.

Response to Amendment

2. The rejection of claims 27, 28, 40 and 45-52 under 35 U.S.C. § 112, second paragraph (see pages 4-6 of the decision) is withdrawn in response to Applicant's amendment. However, a new ground of rejection is set forth below.

The examiner points out that the language of claim 45 does not limit the claim to the embodiment illustrated in Figure 7 (see Applicant's remarks, pages 9-11). In other words, a broadest reasonable interpretation of claim 45 is that the each group of "first" and "second" elements recited in the claim represents the same element at first and second points in time (see pages 5-6 of the decision).

Response to Arguments

3. The examiner does not agree with Applicant's conclusion that the pending claims are allowable (remarks, page 12). Applicant notes that the prior art rejections of the claims were reversed (remarks, pages 7-8). However, the Board further states (see page 4 of the decision):

> It should be understood, however, that our decision in this regard is based solely on the indefiniteness of the claimed subject matter and does not reflect the adequacy or the inadequacy of the prior art evidence applied in support of the rejection before us. Once definite claims are presented, the Examiner is free to apply the same, different, or additional prior art if the Examiner so chooses.

Accordingly, prior art rejections of the pending claims are set forth below.

Claim Rejections under 35 U.S.C. § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. § 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 45-52 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that Applicant regards as the invention.

With respect to claim 45 (currently amended), the claim recites "a coincidence between said address and said first bug address" and "a coincidence between said address and said second bug address." Here, the antecedent basis for the term "said address" is unclear. The examiner presumes that Applicant intends "said address" to refer back to the recited "program address."

With respect to claims 46-52 (previously presented), the claims are dependent on claim 45 and are therefore indefinite for at least the same reason(s) as noted above.

Claim Rejections under 35 U.S.C. § 103

6. The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 27, 28 and 40 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,454,100 to Sagane (already of record, "Sagane") in view of U.S. Patent No. 5,784,537 to Suzuki et al. (already of record, "Suzuki").

With respect to claim 27 (currently amended), Sagane teaches a data processing apparatus (see, for example, electronic apparatus 1 in FIG. 1) comprising:

a bug address setting register adapted to store a bug address, said bug address indicating a start address for a buggy part of a program (see, for example, interrupt generating address register 9 in FIG. 1 and column 3, lines 40-43, which shows that the register stores a correction or bug address, and column 3, lines 32-39, which shows that the correction address indicates a starting address for a buggy part of a program);

a coincidence detecting circuit adapted to compare a program address with said bug address and output an interrupt request signal, said interrupt request signal indicating a coincidence between said program address and said bug address (see, for example, comparator 8 in FIG. 1 and column 3, lines 48-52, which shows that the comparator compares the execution address with the correction address and outputs a coincidence signal, and column 3, lines 59-61, which shows that the coincidence signal is output as an interrupt request signal, and see, for example, column 5, lines 11-16, which further shows that the interrupt request signal indicates a coincidence between the addresses);

a central processing unit adapted to process an interrupt function upon receipt of said interrupt request signal (see, for example, CPU 2 and interrupt control circuit 7d in FIG. 1, which shows that the central processing unit receives the interrupt request signal from the interrupt

control circuit, and column 5, lines 16-21, which shows processing an interrupt function upon receipt of the interrupt request signal).

To the extent Sagane does not explicitly describe that the central processing unit *per se* processes the interrupt function, such an implementation would have been obvious to those of ordinary skill in the art. A person of ordinary skill in the art could, with predictable results, implement the teachings of Sagane such that the central processing unit processes the interrupt function upon receipt of the interrupt request signal.

For example, in an analogous art, Suzuki teaches a PC comparison register section 20 that generates a ROM correction interruption request to a CPU 14 indicating coincidence between the program address and a correction execution interruption address (see, for example, FIG. 1 and column 6, lines 27-33). Suzuki further teaches that the CPU 14 executes interruption processing (i.e., an interrupt function) upon receipt of the ROM correction interruption request (see, for example, column 6, lines 34-36).

Therefore, as Suzuki suggests, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane such that the central processing unit is adapted to process an interrupt function upon receipt of said interrupt request signal.

Sagane does not explicitly describe:

a counter register adapted to store a value, said value representing a number of times said interrupt request signal indicates said coincidence.

Nonetheless, Sagane teaches that the interrupt request signal is generated a number of times (see, for example, steps S7 and S8 in FIG. 2). Sagane further describes that if the ROM 3

includes a plurality of buggy parts, then the interrupt generating address register 9 is updated after each bug to reflect the next correction address (see, for example, column 5, lines 49-54). Thus, the data processing apparatus of Sagane must identify the next correction address from a plurality of correction addresses. To that end, it would have been obvious to those of ordinary skill in the art to track the number of times the interrupt request signal is generated, such as with a counter register adapted to store a value.

For example, Suzuki teaches storing a value in the two most significant bits of a register that represents the number of correcting portions (i.e., the number of buggy parts) in the program (see, for example, FIG. 2B and column 4, lines 41-49). Suzuki describes storing the value and updating the stored value each time the ROM correction interruption request is generated (see, for example, step S5 in FIG. 4A and step S28 in FIG. 4B). Thus, a person of ordinary skill in the art could, with predictable results, derive from the stored value the number of times the ROM correction interruption request was generated.

Therefore, as Sagane and Suzuki suggest, it would have been obvious to those of ordinary skill in the art at the time the invention was made to incorporate within the data processing apparatus of Sagane, a counter register adapted to store a value, said value representing a number of times said interrupt request signal indicates said coincidence.

With respect to claim 28 (previously presented), the rejection of claim 27 is incorporated, and Sagane in view of Suzuki further teaches or suggests that said value is incremented when said interrupt request signal indicates said coincidence.

Suzuki describes that the stored value is decremented when the ROM correction interruption request indicates the coincidence (see, for example, step S28 in FIG. 4B).

Nonetheless, a person of ordinary skill in the art could, with predictable results, implement the teachings of Suzuki such that the stored value is incremented rather than decremented. Indeed, as Suzuki demonstrates, incrementing the value of a counter is within the level of ordinary skill in the art (see, for example, step S88 in FIG. 12B and column 9, lines 52-54).

Specifically, Suzuki teaches storing a value that represents the number of correcting portions S (step S5 in FIG. 4A). Suzuki further illustrates that during the "processing to set [the] ROM correction data for [the] next correcting portion," the stored value is decremented (step S28 in FIG. 4B). Then, the stored value is checked to identify whether the number of correcting portions (i.e., the number of buggy parts) is equal to 0 (step S29 in FIG. 4B). Suzuki states, "If the number of correcting portions S is 0, the processing goes to step S33 since there is no residual correcting portion in the subroutine module of code No. m … [and] if the number of correcting portions S is not 0, the processing goes to step S30 since there [are] still residual correcting portions" left to process (see column 6, line 65 to column 7, line 2).

A person of ordinary skill in the art could implement the teachings of Suzuki such that the stored value is incremented rather than decremented and provide the same result. For example, referring to FIGS. 4A and 4B of Suzuki, given the number of correcting portions S, a person of ordinary skill in the art could initialize the value to 0 (rather than S) in step S5, increment (rather than decrement) the stored value in step S28, and check whether the stored value is equal to S (rather than 0) in step S29. Such an implementation is analogous to the one described in Suzuki and provides the intended result.

Therefore, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane and Suzuki such that said value is incremented when said interrupt request signal indicates said coincidence.

With respect to claim 40 (currently amended), Sagane teaches a data processing apparatus (see, for example, electronic apparatus 1 in FIG. 1) comprising:

program memory adapted to store instruction codes as a program, a program address indicating a location within said program memory for one of the instruction codes (see, for example, ROM 3 in FIG. 1 and column 1, lines 9-14, which shows that the ROM is program memory that stores instruction codes as a program, and column 3, lines 48-52, which shows that an execution or program address indicates a location in the ROM);

a bug address setting register adapted to store a bug address, said bug address indicating a starting address within said program memory for a buggy part of said program (see, for example, interrupt generating address register 9 in FIG. 1 and column 3, lines 40-43, which shows that the register stores a correction or bug address, and column 3, lines 32-39, which shows that the correction address indicates a starting address within the ROM of a buggy part of the program).

Sagane further teaches an interrupt request signal that indicates a coincidence between said program address and said bug address (see, for example, column 3, lines 59-61, which shows an interrupt request signal, and column 3, lines 48-52, which shows that the interrupt request signal indicates a coincidence between the execution address and the correction address), but does not explicitly describe:

a counter register adapted to store a value, said value being incremented when an interrupt request signal indicates a coincidence between said program address and said bug address.

Nonetheless, Sagane teaches that the interrupt request signal is generated a number of times (see, for example, steps S7 and S8 in FIG. 2). Sagane further describes that if the ROM 3 includes a plurality of buggy parts, then the interrupt generating address register 9 is updated after each bug to reflect the next correction address (see, for example, column 5, lines 49-54). Thus, the data processing apparatus of Sagane must identify the next correction address from a plurality of correction addresses. To that end, it would have been obvious to those of ordinary skill in the art to track the number of times the interrupt request signal is generated, such as with a counter register adapted to store a value.

For example, in an analogous art, Suzuki teaches a PC comparison register section 20 that generates a ROM correction interruption request to a CPU 14 indicating coincidence between the program address and a correction execution interruption address (see, for example, FIG. 1 and column 6, lines 27-33). Suzuki further teaches storing a value in the two most significant bits of a register that represents the number of correcting portions (i.e., the number of buggy parts) in the program (see, for example, FIG. 2B and column 4, lines 41-49). Suzuki describes storing the value and updating the stored value each time the ROM correction interruption request is generated (see, for example, step S5 in FIG. 4A and step S28 in FIG. 4B). Thus, a person of ordinary skill in the art could, with predictable results, derive from the stored value the number of times the ROM correction interruption request was generated.

Therefore, as Sagane and Suzuki suggest, it would have been obvious to those of ordinary skill in the art at the time the invention was made to incorporate within the data processing apparatus of Sagane, a counter register adapted to store a value. Sagane in view of Suzuki further teaches or suggests said value being incremented when an interrupt request signal indicates a coincidence between said program address and said bug address.

Suzuki describes that the stored value is decremented when the ROM correction interruption request indicates the coincidence (see, for example, step S28 in FIG. 4B). Nonetheless, a person of ordinary skill in the art could, with predictable results, implement the teachings of Suzuki such that the stored value is incremented rather than decremented. Indeed, as Suzuki demonstrates, incrementing the value of a counter is within the level of ordinary skill in the art (see, for example, step S88 in FIG. 12B and column 9, lines 52-54).

Specifically, Suzuki teaches storing a value that represents the number of correcting portions S (step S5 in FIG. 4A). Suzuki further illustrates that during the "processing to set [the] ROM correction data for [the] next correcting portion," the stored value is decremented (step S28 in FIG. 4B). Then, the stored value is checked to identify whether the number of correcting portions (i.e., the number of buggy parts) is equal to 0 (step S29 in FIG. 4B). Suzuki states, "If the number of correcting portions S is 0, the processing goes to step S33 since there is no residual correcting portion in the subroutine module of code No. m … [and] if the number of correcting portions S is not 0, the processing goes to step S30 since there [are] still residual correcting portions" left to process (see column 6, line 65 to column 7, line 2).

A person of ordinary skill in the art could implement the teachings of Suzuki such that the stored value is incremented rather than decremented and provide the same result. For

example, referring to FIGS. 4A and 4B of Suzuki, given the number of correcting portions S, a person of ordinary skill in the art could initialize the value to 0 (rather than S) in step S5, increment (rather than decrement) the stored value in step S28, and check whether the stored value is equal to S (rather than 0) in step S29. Such an implementation is analogous to the one described in Suzuki and provides the intended result.

Therefore, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane and Suzuki such that said value is incremented when an interrupt request signal indicates a coincidence between said program address and said bug address.

Sagane in view of Suzuki further teaches or suggests:

wherein another program address indicates a location within said program memory for another of the instruction codes (see, for example, Sagane, column 3, lines 48-52, which shows that another execution or program address indicates another location in the ROM).

8. Claims 45-50 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Sagane in view of Suzuki and in view of U.S. Patent No. 6,412,081 to Koscal et al. (already of record, "Koscal").

With respect to claim 45 (currently amended), Sagane teaches a data processing apparatus (see, for example, electronic apparatus 1 in FIG. 1) comprising:

program memory adapted to store instruction codes as a program, a program address indicating a location within said program memory for one of the instruction codes (see, for example, ROM 3 in FIG. 1 and column 1, lines 9-14, which shows that the ROM is program

memory that stores instruction codes as a program, and column 3, lines 48-52, which shows that an execution or program address indicates a location in the ROM);

a central processing unit adapted to process interrupt functions, one of said interrupt functions being processed upon receipt of a first interrupt request signal or a second interrupt request signal (see, for example, CPU 2 and interrupt control circuit 7d in FIG. 1, which shows that the central processing unit receives an interrupt request signal from the interrupt control circuit, and column 5, lines 16-21, which shows processing an interrupt function upon receipt of the interrupt request signal).

To the extent Sagane does not explicitly describe that the central processing unit *per se* processes the interrupt functions, such an implementation would have been obvious to those of ordinary skill in the art. A person of ordinary skill in the art could, with predictable results, implement the teachings of Sagane such that the central processing unit processes one of the interrupt functions upon receipt of the interrupt request signal.

For example, in an analogous art, Suzuki teaches a PC comparison register section 20 that generates a ROM correction interruption request to a CPU 14 indicating coincidence between the program address and a correction execution interruption address (see, for example, FIG. 1 and column 6, lines 27-33). Suzuki further teaches that the CPU 14 executes interruption processing (i.e., an interrupt function) upon receipt of the ROM correction interruption request (see, for example, column 6, lines 34-36).

Therefore, as Suzuki suggests, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane such that the central processing unit is adapted to process interrupt functions, one of said

interrupt functions being processed upon receipt of a first interrupt request signal or a second interrupt request signal.

Sagane and Suzuki do not explicitly describe that the central processing unit is adapted to process interrupt functions of different priority levels.

Nonetheless, central processing units are known in the art to process interrupt functions of different priority levels. For example, in an analogous art, Koscal teaches processing interrupt functions to patch bugs in a program (see, for example, the abstract). Koscal further teaches that the microprocessor is adapted to execute interrupt functions of different priority levels, so as to provide for immediate execution of interrupt functions with higher priority levels (see, for example, column 14, lines 2-7).

Therefore, as Koscal suggests, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane and Suzuki such that the central processing unit is adapted to process interrupt functions of different priority levels.

Sagane in view of Suzuki and Koscal further teaches or suggests:

a first coincidence detecting circuit adapted to compare said program address with a first bug address and output said first interrupt request signal, said central processing unit receiving said first interrupt request signal (see, for example, Sagane, comparator 8 in FIG. 1 and column 3, lines 48-52, which shows that the comparator compares the execution address with a correction or bug address and outputs a coincidence signal, and column 3, lines 59-61, which shows that the coincidence signal is output as an interrupt request signal, and see, for example, column 5, lines 11-16, which further shows that the interrupt request signal indicates a

coincidence between the addresses, and FIG. 1, which shows that the central processing unit receives the interrupt request signal from the interrupt control circuit);

a second coincidence detecting circuit adapted to compare said program address with a second bug address and output said second interrupt request signal, said central processing unit receiving said second interrupt request signal (see, for example, Sagane, comparator 8 in FIG. 1 and column 3, lines 48-52, which shows that the comparator compares the execution address with a correction or bug address and outputs a coincidence signal, and column 3, lines 59-61, which shows that the coincidence signal is output as an interrupt request signal, and see, for example, column 5, lines 11-16, which further shows that the interrupt request signal indicates a coincidence between the addresses, and FIG. 1, which shows that the central processing unit receives the interrupt request signal from the interrupt control circuit).

Sagane describes that if the ROM 3 includes a plurality of buggy parts, then the interrupt generating address register 9 is updated after each bug to reflect the next correction address (see, for example, column 5, lines 49-54). In an analogous embodiment, Sagane likewise teaches a correction address register 21 that is updated to reflect the next correction address (see, for example, column 6, lines 63-67), and further suggests, as an alternative, providing a plurality of correction address registers 21 and a plurality of comparators 8 for the plurality of buggy parts (see, for example, column 6, line 67 to column 7, line 3). Thus, a person of ordinary skill in the art could, with predictable results, incorporate a plurality of comparators (i.e., coincidence detecting circuits) into the data processing apparatus of Sagane.

Therefore, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane, Suzuki and Koscal

such that the data processing apparatus comprises first and second coincidence detecting circuits adapted to compare said program address with first and second bug addresses and output said first and second interrupt request signals, respectively.

Sagane further teaches that the interrupt request signal indicates a coincidence between said program address and said first bug address or a coincidence between said program address and said second bug address (see, for example, column 3, lines 48-52, which shows that the interrupt request signal indicates a coincidence between the execution address and the correction address), but Sagane, Suzuki and Koscal do not explicitly describe:

a counter register adapted to store a value, said value being incremented by 1 when said first interrupt request signal indicates a coincidence between said address and said first bug address or when said second interrupt request signal indicates a coincidence between said address and said second bug address,

wherein said counter register is set to 0 during initialization processing.

Nonetheless, Sagane teaches that the interrupt request signal is generated a number of times (see, for example, steps S7 and S8 in FIG. 2). As noted above, Sagane further describes that if the ROM 3 includes a plurality of buggy parts, then the interrupt generating address register 9 is updated after each bug to reflect the next correction address (see, for example, column 5, lines 49-54). Thus, the data processing apparatus of Sagane must identify the next correction address from a plurality of correction addresses. To that end, it would have been obvious to those of ordinary skill in the art to track the number of times the interrupt request signal is generated, such as with a counter register adapted to store a value.

For example, Suzuki teaches storing a value in the two most significant bits of a register that represents the number of correcting portions (i.e., the number of buggy parts) in the program (see, for example, FIG. 2B and column 4, lines 41-49). Suzuki describes storing the value and updating the stored value each time the ROM correction interruption request is generated (see, for example, step S5 in FIG. 4A and step S28 in FIG. 4B). Thus, a person of ordinary skill in the art could, with predictable results, derive from the stored value the number of times the ROM correction interruption request was generated.

Therefore, as Sagane and Suzuki suggest, it would have been obvious to those of ordinary skill in the art at the time the invention was made to incorporate within the data processing apparatus of Sagane, a counter register adapted to store a value. Sagane in view of Suzuki further teaches or suggests said value being incremented by 1 when said first interrupt request signal indicates a coincidence between said address and said first bug address or when said second interrupt request signal indicates a coincidence between said address and said second bug address, wherein said counter register is set to 0 during initialization processing.

Suzuki describes that the stored value is decremented when the ROM correction interruption request indicates the coincidence (see, for example, step S28 in FIG. 4B). Nonetheless, a person of ordinary skill in the art could, with predictable results, implement the teachings of Suzuki such that the stored value is incremented rather than decremented. Indeed, as Suzuki demonstrates, incrementing the value of a counter is within the level of ordinary skill in the art (see, for example, step S88 in FIG. 12B and column 9, lines 52-54).

Specifically, Suzuki teaches storing a value that represents the number of correcting portions S (step S5 in FIG. 4A). Suzuki further illustrates that during the "processing to set [the]

ROM correction data for [the] next correcting portion," the stored value is decremented (step S28 in FIG. 4B). Then, the stored value is checked to identify whether the number of correcting portions (i.e., the number of buggy parts) is equal to 0 (step S29 in FIG. 4B). Suzuki states, "If the number of correcting portions S is 0, the processing goes to step S33 since there is no residual correcting portion in the subroutine module of code No. m … [and] if the number of correcting portions S is not 0, the processing goes to step S30 since there [are] still residual correcting portions" left to process (see column 6, line 65 to column 7, line 2).

A person of ordinary skill in the art could implement the teachings of Suzuki such that the stored value is incremented rather than decremented and provide the same result. For example, referring to FIGS. 4A and 4B of Suzuki, given the number of correcting portions S, a person of ordinary skill in the art could initialize the value to 0 (rather than S) in step S5, increment (rather than decrement) the stored value in step S28, and check whether the stored value is equal to S (rather than 0) in step S29. Such an implementation is analogous to the one described in Suzuki and provides the intended result.

Therefore, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane, Suzuki and Koscal such that said value is incremented by 1 when said first interrupt request signal indicates a coincidence between said address and said first bug address or when said second interrupt request signal indicates a coincidence between said address and said second bug address, wherein said counter register is set to 0 during initialization processing.

With respect to claim 46 (previously presented), the rejection of claim 45 is incorporated, and Sagane in view of Suzuki and Koscal further teaches or suggests that said counter register is

located within a random access memory at a predetermined memory address (see, for example, Suzuki, column 5, lines 41-46, which shows that the value is stored at a predetermined location within RAM).

With respect to claim 47 (previously presented), the rejection of claim 45 is incorporated, and Sagane in view of Suzuki and Koscal further teaches or suggests:

bug address setting registers adapted to store said first and second bug addresses (see, for example, Sagane, interrupt generating address register 9 in FIG. 1 and column 3, lines 40-43, which shows that the register stores the correction or bug address).

With respect to claim 48 (previously presented), the rejection of claim 45 is incorporated, and Sagane in view of Suzuki and Koscal further teaches or suggests that said first bug address indicates a starting address within said program memory for a first buggy part of said program, and said second bug address indicates a starting address within said program memory for a second buggy part of said program (see, for example, Sagane, column 3, lines 32-39, which shows that that the correction or bug addresses indicate starting addresses within the ROM of buggy parts of the program).

With respect to claim 49 (previously presented), the rejection of claim 48 is incorporated, and Sagane in view of Suzuki and Koscal further teaches or suggests that said central processing unit is adapted use said value to select for correction said first buggy part or said second buggy part (see, for example, Suzuki, column 6, line 65 to column 7, line 7, which shows that the value is used to select the corresponding buggy part for correction).

With respect to claim 50 (previously presented), the rejection of claim 45 is incorporated, and Sagane in view of Suzuki and Koscal further teaches or suggests that said first and second interrupt request signals are input to said central processing unit as two different interrupt request signals (see, for example, Sagane, column 5, lines 49-54, which shows that the interrupt request signal is generated separately for each buggy part of the program, and Suzuki, column 6, lines 27-36, which further shows that the interrupt request signals are input to the CPU).

9. Claims 51 and 52 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Sagane in view of Suzuki and in view of Koscal, as applied to claim 45 above, and further in view of U.S. Patent No. 5,701,506 to Hosotani (already of record, "Hosotani").

With respect to claim 51 (previously presented), the rejection of claim 45 is incorporated, and Sagane in view of Suzuki and Koscal further teaches or suggests that said first and second interrupt request signals are input to said central processing unit (see, for example, Suzuki, column 6, lines 27-36), but does not explicitly describe that said first and second interrupt request signals are input to said central processing unit as a single interruption.

However, in an analogous art, Hosotani teaches a plurality of coincidence detecting circuits that each output a signal indicating a coincidence between a program address and a bug address (see, for example, match circuits 9 in FIG. 2 and column 4, lines 30-59). Hosotani further teaches that the coincidence signals are combined into a single signal (see, for example, column 4, line 60 to column 5, line 2). The teachings of Hosotani provide for correcting a plurality of bugs in a program stored in a ROM without remaking the ROM (see, for example, column 1, lines 45-50).

Therefore, as Hosotani suggests, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane, Suzuki and Koscal such that said first and second interrupt request signals are input to said central processing unit as a single interruption, thus providing an optimized mechanism for correcting a plurality of bugs in a program.

With respect to claim 52 (previously presented), the rejection of claim 51 is incorporated, and Sagane in view of Suzuki, Koscal and Hosotani further teaches or suggests that said first and second interrupt request signals are AND'ed together to become said single interruption.

Hosotani teaches that the coincidence signals are OR'ed together, and that the resulting output is at a "1" level when any one of the address comparisons is a match and at a "0" level when all of the comparisons are mismatches (see, for example, column 4, line 60 to column 5, line 2). In other words, Hosotani defines the coincidence detection mechanism as "active high." When the mechanism is instead defined as "active low," a person of ordinary skill in art could, with predictable results, substitute the OR gate 14 with an AND gate to provide the same result. The resulting output from the AND gate, in such a case, would be at a "0" level when any one of the address comparisons is a match and at a "1" level when all of the comparisons are mismatches. The truth tables below illustrate:

Active high (a "1" level represents a match or coincidence between the addresses)

First signal (S_1)	Second signal (S_2)	S_1 OR S_2	Result
0	0	0	No Interrupt
0	1	1	Interrupt
1	0	1	Interrupt
1	1	1	Interrupt

Active low (a "0" level represents a match or coincidence between the addresses)

First signal (S_1)	Second signal (S_2)	S_1 AND S_2	Result
1	1	1	No Interrupt
1	0	0	Interrupt
0	1	0	Interrupt
0	0	0	Interrupt

Therefore, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the data processing apparatus of Sagane, Suzuki, Koscal and Hosotani such that said first and second interrupt request signals are AND'ed together to become said single interruption.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure (see the attached Notice of References Cited).

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday to Friday from 9:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Michael J. Yigdall/
Primary Examiner, Art Unit 2192